Let's look at solutions to the interclass problem.
What did we talk about last class?
Do you have any questions about the reading?
As odd at it might sound, the original versions of Java did not include a way to do simple text input. This is because few modern applications, especially in the fields Java was aimed at, actually do simple text input.

With Java 5 the java.util.Scanner class was added. Largely at the urging of educators who used Java and wanted simple text input.

Let's go look at java.util.Scanner in the API to see what it can do, then write a simple program that uses it.
In PAD1 one of the primary skills you had to learn was functional decomposition, how to break a problem into different functions.

In an OO environment like Java, you have to think at a different level and learn how to decompose your problem into classes. Then you must think about what those classes need to store and what they can do.
To demonstrate this process I want us to play with one of my favorite problems: a ray tracer.

Let's go through and list some of the classes that we might need to begin writing a ray tracer. Then we'll start filling them in and derive some of the math for doing intersections of rays with some geometric primitives.
How do you think you should go about picking the classes you need to solve a particular problem?

The design for the first assignment is due a week from today. That means you will have to be able to describe the game you want to write by then.

Interclass problem – See how close you can come to making a text based game of tic-tac-toe. You can make a 3x3 array of chars with the following code.

- char[][] board=new char[3][3];